Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Aleks Kobentar

# Vizualizacija imen novorojenčkov v povezavi s podatki iz baze IMDb in Wikipedije

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: viš. pred. dr. Alenka Kavčič
Somentor: as. dr. Matevž Pesek

Ljubljana, 2019

*Besedilo je oblikovano z urejevalnikom besedil LaTeX.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:
Vizualizacija imen novorojenčkov v povezavi s podatki iz baze IMDb in Wikipedije

Tematika naloge:

Statistični urad Republike Slovenije beleži podatke o imenih novorojenih otrok v Sloveniji ter jih kot odprte podatke javnega sektorja nudi tudi širši javnosti. Podatki so ločeni po spolu in posameznih letih. Podatki o imenih so zanimivi predvsem v kombinaciji s podrobnejšimi podatki o posameznem imenu, njegovem izvoru in podobnih oz. izpeljanih imenih, pa tudi v navezavi s slavnimi osebnostmi (npr. iz filmskega sveta) s tem imenom. V okviru diplomske naloge zasnujte in izdelajte spletno aplikacijo za vizualizacijo podatkov o imenih novorojenčkov v Sloveniji. Aplikacija naj uporabniku omogoča prikaz najpogostejših imen glede na leto rojstva in spol, iskanje imen, ki ustrezajo določenim kriterijem (npr. glede vsebovanih črk ali dolžine imena), za izbrano ime pa naj prikaže tudi podrobnejše podatke, ki jih pridobite z Wikipedije, in podatke o znanih igralcih s tem imenom, ki jih pridobite iz baze IMDb. Vse vizualne predstavitve naj bodo izbrane tako, da bodo ustrezale vrsti prikazanih podatkov, hkrati pa morajo omogočati enostavno in intuitivno interakcijo in izbiro kriterijev za prikaz.

*Za pomoč in usmerjanje pri diplomski nalogi se zahvaljujem mentorici viš. pred. dr. Alenki Kavčič in somentorju as. dr. Matevžu Pesku.*

# Kazalo

# Seznam uporabljenih kratic

| kratica | angleško | slovensko |
|---|---|---|
| **CORS** | Cross-Origin Resource Sharing | Navzkrižna delitev podatkov |
| **IMDb** | Internet Movie Databasse | Spletna podatkovna zbirka filmov |
| **BDP** | Gross domestic product | Bruto domači proizvod |
| **JSON** | JavaScript Object Notation | Opis JavaScript objekta |
| **URL** | Uniform Resource Locator | Spletni naslov |

# Povzetek

**Naslov:** Vizualizacija imen novorojenčkov v povezavi s podatki iz baze IMDb in Wikipedije

**Avtor:** Aleks Kobentar

Diplomska naloga opisuje postopek pridobivanja in prikaza podatkov iz različnih virov. Vir podatkov predstavljajo tri podatkovne zbirke. S Statističnega urada Republike Slovenije so podatki o številu imen novorojenčkov od leta 1992 do 2017. IMDb-podatkovna zbirka nudi podatke o igralcih in filmih. Zadnji vir podatkov je spletna enciklopedija Wikipedija, ki nudi zanimive podatke o podrobnostih imen.

Združitev vseh podatkov zahteva uporabo različnih orodij. Za uvoz podatkov je uporabljen programski jezik Python. Za hranjenje podatkov o številu imen novorojenčkov je uporabljena nerelacijska podatkovna zbirka Elasticsearch. Za potrebe vizualizacije podatkov, ki so shranjeni na spletnih straneh ali v lokalni podatkovni zbirki, so implementirani strežniki s programskima jezikoma Python in Node.js. Poleg osnovnih spletnih orodij sta JavaScript in knjižnica D3.js uporabljena kot glavno orodje za prikaz podatkov.

**Ključne besede:** spletni strežnik, podatkovna zbirka, vir podatkov, vizualizacija, imena otrok.

# Abstract

**Title:** Visualization of baby names combined with IMDb and Wikipedia data

**Author:** Aleks Kobentar

This Bachelor's Thesis describes the process of collecting and visualizing data from different sources. There are three different data sources. The first source is from the Statistical office of Slovenia where there is data about the number of baby names occurring from 1992 to 2017. The second sourse is the IMDb database, which has data about actors and movies. The third data source is the free Wikipedia encyclopedia , which holds interesting data about names.

To be able to merge all the datasources requires a great range of frameworks. For importing the data, the programming language Python is used. For data storage about the number of babynames, the unrelation database Elasticsearch is used. For the exchange of data which is stored on the internet or on local machine servers, either Python or Node.js. are implemented. In addition, the basic web technologies JavaScript and D3.js are the main tools for data visualization.

**Keywords:** web server, database, source of data, visualization, baby names.

# Poglavje 1

# Uvod

Zbiranje podatkov srečamo na vsakem koraku, čeprav se mogoče nekateri tega še ne zavedajo. Drugim zbiranje podatkov že povzroča skrbi. Že ob našem rojstvu se zabeležijo podatki na Statističnem uradu Republike Slovenije, saj vodijo podatke o številu imen in priimkov. Poleg omenjenih hranijo podatke še za veliko ostalih kategorij, npr. za: BDP, cene in inflacijo, delo in brezposelnost, izobraževanje.

Vsi podatki Statističnega urada Republike Slovenije so javno dostopni. Za nekatere kategorije podatkov je vizualizacija implementirana, za druge žal ne obstaja. SURS se zaveda, da podatki utegnejo zanimati širšo javnost. Zaradi tega so v zadnjem času začeli z vizualizacijo podatkov [19].

Za podatke o imenih se zanimajo predvsem starši pri izbiri imena za svojega otroka in redki posamezniki, ki jih zanimajo imena. Izziv diplomske naloge je vizualizacija podatkov imen novorojenčkov.

Tabelaričen prikaz podatkov tipično ni privlačen, zato z diplomsko nalogo želimo podatke predstaviti širši javnosti z uporabo vizualizacije. Iskanje po tabeli je zamudno in dolgočasno. Da izvemo poljuben podatek, se moramo sprehoditi čez celotno tabelo. Enako velja v primeru, če želimo izvedeti, katerega leta je bilo izbrano ime najbolj pogosto. Z vizualizacijo želimo na takšna in podobna vprašanja odgovoriti hitro in intuitivno.

Poleg števila rojstev želimo o izbranem imenu izvedeti čim več. Z uporabo

spleta najdemo najrazličnejše podatke o imenu, a nam brskanje po spletu vzame veliko časa, zato želimo uporabniku na enem mestu prikazati čim več zanimivih in uporabnih informacij.

Izziv diplomske naloge vidim v zahtevnosti vizualizacije in implementacije. Vizualizacija zahteva veliko občutka za estetiko, implementacija v ozadju pa zahteva znanje HTTP-protokola [4], programskih jezikov Python [20] in JavaScript [6], knjižnice D3.js [1] in operacijskega sistema Linux [8].

V okviru diplomske naloge so v drugem poglavju predstavjene tehnologije, ki smo jih uporabili pri izdelavi diplomske naloge. Poglevje 3 opisuje strukturo, pridobivanje in shranjevanje podatkov. Poglavje 4 opisuje problem navzkrižne delitve podatkov in rešitev za ta problem. Poglavje 5 opisuje aplikacijo. V zadnjem 6. poglavju je povzetek diplomske naloge.

# Poglavje 2

# Uporabljene tehnologije

Izdelava spletne vizualizacije zahteva veliko količino različnih podatkov. Za shranjevanje in prenos podatkov torej potrebujemo tehnologije, ki nam to omogočajo. V tem poglavju so predstavljene ključne tehnologije, uporabljene za izdelavo vizualizacije. Nekatere osnovne tehnologije za oblikovanje strani so izpuščene.

## 2.1 Elasticsearch

Elasticsearch [3] je trenutno eden najbolj hitrorastočih porazdeljenih iskalnikov ta hip. Razvit je v programskem jeziku Java in je odprtokoden. Za dostop je razvitih veliko odjemalcev, za programske jezike Java, NET, PHP, Python, Ruby in druge. Elasticsearch je distribuiran sistem, kar omogoča porazdelitev podatkov po več sistemih (cluster). Koordinacija med distribuiranimi podatki se izvede avtomatično. Poleg same hrambe in iskanja podatkov obstaja veliko orodij, ki Elasticsearch naredijo še bolj zanimiv, a jih v obsegu diplomske naloge ne bomo obravnavali. Vseeno omenimo Kibano, orodje, ki je namenjeno predvsem preprosti vizualizaciji podatkov brez znanja programiranja.

Elasticsearch kot enoto zapisa uporablja tako imenovani dokument (document), sestavljen iz poljubnih ključev in vrednosti. Vsi podatki enega

dokumenta so shranjeni v JSON-obliki. Enake dokumente združimo v tako imenovane indekse (index). Glavna značilnost dokumenta je nepredpisana struktura, kar pomeni, da vsak dokument lahko vsebuje poljubne ključe in vrednosti.

## 2.2 Programski jezik Python

Programski jezik Python smo uporabili za implementacijo vmesnika med podatkovno zbirko IMDb in glavnim strežnikom, napisanem v Node.js, ki glede na zahtevo spletne strani oz. uporabnika izvede poizvedbo na strežnik IMDb. Pridobljene podatke ustrezno preoblikuje v JSON in jih preko HTTP-protokola na zahtevo pošlje spletni aplikaciji.

## 2.3 Operacijski sistem Linux

Nastavitve okolja, v katerem delamo, so zelo pomembne. Nezavedanje nastavitev lahko kasneje povzroči veliko težav. Spletni strežnik teče na operacijskem sistemu Linux. V slovenščini uporabljamo šumnike, zato so nastavitve terminalnega okna zelo pomembne. Z ukazom `locale -at` lahko vidimo vse nastavitve trenutnega okolja [9]. Nastavitve okolja uporablja Linux za predstavitev teksta, regij, časa in podobnega. Za nastavitve vrednosti se uporablja ukaz `export` [10]. V mojem primeru sem nastavil dve spremenljivki `LC_ALL` in `LC_CTYPE` na vrednosti `en_US.UTF8`. Za nastavitev izvedemo ukaza `export LC_ALL='en_US.UTF8'` in `export LC_CTYPE='en_US.UTF8'`. Za potrditev nastavitev je potrebno zagnati ukaz `sudo dpkg reconfigure locales`, ki potrdi nastavitve. Brez teh nastavitev namestitev paketa IMDbPY (predstavljen v naslednjih razdelkih) ni bila mogoča.

## 2.4 Node.js

Node.js [12] je odprtokodna rešitev, namenjena implementaciji strežnikov. Podpira operacijske sisteme Windows, Linux, Unix, Mac OS in druge. Temelji na programskem jeziku JavaScript. Glavna prednost uporabe v primerjavi s podobnimi rešitvami, kot so PHP ali ASP, je v tem, da ne čaka ostalih operacij, da se zaključijo, ampak teče v asinhronem načinu. Node.js omogoča generiranje dinamičnih spletnih strani, kar pomeni, da lahko del procesiranja spletne strani izvedemo na strani strežnika. Omogoča ustvarjanje, branje, pisanje, brisanje in zapiranje dokumentov [13].

# Poglavje 3

# Struktura in shranjevanje podatkov

V tem poglavju je predstavljena struktura izvornih podatkov. Aplikacija črpa podatke iz treh različnih virov. Glavni vir podatkov predstavlja Statistični urad Republike Slovenije. Od tam se prenesejo podatki o številu rojstev skozi časovno obdobje in o tem, kolikšno je skupno število prebivalstva s tem imenom [14, 15]. Drugi vir podatkov predstavlja največja podatkovna zbirka filmske industrije IMDb [16]. Od tam se prenesejo podatki o filmskih igralcih in njihovih filmih. Zadnji vir podatkov predstavlja največja prosta spletna tehnologija Wikipedija [23]. S spletne strani se prenesejo različni zanimivi podatki za iskano ime.

## 3.1 Statistični urad Republike Slovenije

Statistični urad Republike Slovenije (SURS) shranjuje različne podatke. Poleg že omenjenih podatkov o številu imen novorojenčkov skozi čas bomo za namen diplomske naloge s spletne strani za daljše časovno obdobje pridobili še podatke o tem, koliko prebivalcev Slovenije nosi iskano ime.

### 3.1.1 Podatki Statističnega urada Republike Slovenije o številu imen novorojenčkov skozi čas

Podatki o imenih novorojenčkov so razdeljeni v dve datoteki na dečke [15] in deklice [14]. Dostopni so v več oblikah, a za potrebe diplomske naloge smo uporabili razmejeno obliko brez glave v formatu CSV. V vsaki datoteki ena vrstica predstavlja podatke za eno ime. Za ime Aleks imamo naslednji zapis: "Aleks";22;25;22;21;30;26;29;25;47;54;46;39;51;56;46;56;62;68;81;80;82;101; 73;72;86;85;83;76;76;78;65;69;62;71;52;47;52;54;48;43;51;46;45;43;34; 35;34; 25;35;33;30;28

Vsaka vrstica vsebuje tri različne kategorije podatkov. Prvi del predstavlja ime, drugi število pojavitev po letih in tretji rang po letih. Vsakemu imenu torej sledi 26 vrednosti, ki predstavljajo število pojavitev tega imena od leta 1992 do leta 2017. Zadnjih 26 vrednosti predstavlja rang imena med letoma 1992 in 2017. Omenjena vrstica je v tabelarični obliki (glej sliko 3.1).

| | Število | | | | | | | | | | | | | | | | | | | | | | | | | | | Rang | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Leto | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
| Aleks | 22 | 25 | 22 | 21 | 30 | 26 | 29 | 25 | 47 | 54 | 46 | 39 | 51 | 56 | 46 | 56 | 62 | 68 | 81 | 80 | 82 | 101 | 73 | 72 | 86 | 85 | 83 | 76 | 76 | 78 | 65 | 69 | 62 | 71 | 52 | 47 | 52 | 54 | 48 | 43 | 51 | 46 | 45 | 43 | 34 | 35 | 34 | 25 | 35 | 33 | 30 | 28 |

Slika 3.1: Tabelarični prikaz podatkov SURS

#### Težave s šumniki pri slovenskih imenih

Pri slovenskih imenih moramo paziti, saj imena vsebujejo šumnike. Napačno kodiranje šimnikov bi pomenilo, da aplikacije imena ne bi našle. Namesto šumnikov se lahko pojavijo tudi drugi znaki, ki lahko povzročajo težave pri iskanju. Temu se najlažje izognemo, če med vsako manipulacijo z datotekami (premikanje, kopiranje, preimenovanje) pazimo, da je kodiranje vedno nastavljeno na utf-8. Nastavitev kodiranja preverimo z ukazom `file -I imeDatoteke`.

### Shranjevanje podatkov v podatkovno zbirko Elasticsearch

Podatkovna zbirka Statističnega urada Republike Slovenije ni javno dostopna. Podatke o imenih je mogoče pridobiti le v datotekah z različnimi formati: `.px`, `.xls`, `.txt`, `.csv` in `.html`. Za delovanje aplikacije potrebujemo podatke, dostopne na strežniku, zato smo podatke shranili v nerelacijsko podatkovno zbirko Elasticsearch.

Podatki, pridobljeni s Statističnega urada Republike Slovenije, se nahajajo v dveh datotekah. V eni datoteki so zbrana vsa ženska imena, v drugi vsa moška imena. Podatke smo razdelili na manjše dele, imenovane dokumenti. Vsak dokument ima svojo zaporedno številko (id). Podatke razdelimo po principu dveh ključev: ime in leto. Vsak dokument vsebuje zaporedno številko (id), ime, leto, število imen, rang in spol. V Elasticsearchu je zapis predstavljen v obliki JSON (glej sliko 3.2).

Za vsako ime smo ustvarili 26 dokumentov, ker imamo podatke za obdobje šestindvajsetih let. Za uvoz podatkov smo implementirali skripto v jeziku Python. Skripta vsebuje knjižnici Elasticsearch [22] in csv [21]. Z uporabo knjižnice csv skripta prebere vsebino obeh datotek, katerih imeni sta nastavljeni s spremenljivkama `source_m` in `source_z`. Glede na ime datoteke se v spremenljivko `spol` shrani bodisi znak `m`, ki predstavlja moško ime, ali znak `z`, ki predstavlja žensko ime. Skripta prebere vsebino in se sprehodi po vrsticah `csv` datoteke. Vsako vrstico datoteke razdeli na tri dele: ime, seznam števil in seznam rangov. Ime se nahaja na prvem mestu oz. do prvega znaka ";". Ostali del razdeli na dva enako dolga dela. Prvi del je seznam, ki vsebuje število pojavitev imena med 1992 in 2017. Drugi del seznama vsebuje range med letoma 1992 in 2017. Za vsako leto iz seznama v spremenljivko shrani ustrezne vrednosti za število, rang in ime. Vse tri vrednosti skupaj z zaporedno številko dokumenta in spolom pošlje kot argument funkciji, ki iz prejetih argumentov zgradi slovar. Primer slovarja:

```
zapis = {
       'id':  1,
       'ime':  'Aleks',
```

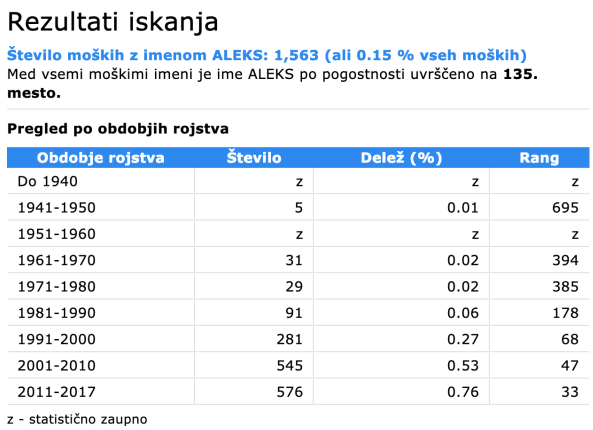

Slika 3.2: Primer dokumenta v obliki JSON

```
        'leto':  1996,
        'stevilo':  10,
        'rang':  10,
        'spol':  'm'
}
```

Slovar kot parameter pošlje funkciji knjižnice Elasticsearch. Funkcija izvede vgrajeno funkcijo Elasticsearch knjižnice z ukazom `res = es.index(index='babynames_test', doc_type='_doc', id=str(record_id), body=zapis)`.

### 3.1.2 Podatki Statističnega urada Republike Slovenije za daljše časovno obdobje pred letom 1992

Podatki za vsako leto so na voljo le od leta 1992 dalje. Pred letom 1992 so podatki združeni v časovna obdobja desetih let. V aplikaciji bomo prikazali prvi odstavek, ki vsebuje seštevek vseh imen za vsa leta (glej sliko 3.3 prvi odstavek (besedilo v modri barvi)). Podatki so predstavljeni v obliki spletne strani oz. v obliki HTML-značk (HTML tag).



Rezultati iskanja

**Število moških z imenom ALEKS: 1,563 (ali 0.15 % vseh moških)**
Med vsemi moškimi imeni je ime ALEKS po pogostnosti uvrščeno na **135. mesto.**

**Pregled po obdobjih rojstva**

| Obdobje rojstva | Število | Delež (%) | Rang |
|---|---|---|---|
| Do 1940 | z | z | z |
| 1941-1950 | 5 | 0.01 | 695 |
| 1951-1960 | z | z | z |
| 1961-1970 | 31 | 0.02 | 394 |
| 1971-1980 | 29 | 0.02 | 385 |
| 1981-1990 | 91 | 0.06 | 178 |
| 1991-2000 | 281 | 0.27 | 68 |
| 2001-2010 | 545 | 0.53 | 47 |
| 2011-2017 | 576 | 0.76 | 33 |

z - statistično zaupno

Slika 3.3: SURS – skupno število pojavitev imena

## 3.2 Podatkovna zbirka IMDb

### 3.2.1 Podatki iz zbirke IMDb

Podatkovna zbirka IMDb (Internet Movie Database) je najbolj popularen vir informacij iz sveta filmov. V podatkovni zbirki je več kot 250 milijonov podatkov, več kot 4 milijone filmov ter več kot 8 milijonov igralcev, režiserjev in drugih članov filmske industrije. Vsak mesec spletno stran obišče 250 milijonov obiskovalcev [24].

Podatkovna zbirka IMDb je javno dostopna na 2 načina, a vsakršna objava je prepovedana. Prvi način dostopa je prenos sedmih datotek s spletne strani [5]. Drugi način je neposredni dostop, ki je opisan v naslednji točki.

### 3.2.2 Pridobivanje podatkov IMDb

#### Knjižnica IMDbPY

Za povezavo s podatkovno zbirko IMDb je najbolje uporabiti knjižnico IMDbPY. Knjižnica je napisana v programskem jezuku Python in je prilagojena povezavi s podatkovno zbirko IMDb. Povezava je mogoča na lokalno podatkovno zbirko ali na javno dostopno podatkovno zbirko. Knjižnica poleg

povezave s podatkovno zbirko vsebuje funkcije za iskanje po različnih kategorijah, kot so filmi in igralci.

### Uporaba knjižnice IMDbPY

V nekaterih primerih nastavitev Python ne vključuje urejevalnika paketov, ki je nujen za namestitev potrebnih knjižnic. Če je nameščen, lahko preverimo z ukazom `pip --version`. PIP lahko namestimo z ukazom `sudo apt install python-pip`. Knjižnico IMDbPY namestimo z ukazom `pip install git+https://github.com/alberanid/imdbpy`. Žal je dokumentacija, ki je na voljo na spletu, precej kratka in v njej ni enostavno najti željenih funkcij [2].

### Vključevanje knjižnice

Prvi korak je vključitev knjižnice IMDbPY in klic funkcije, ki ustvari povezavo s podatkovno zbirko (glej sliko 3.4). V privzetem načinu se povezava ustvari na javno dostopno podatkovno zbirko in konfiguracija ni potrebna.

```
>>> import imdb
>>> ia = imdb.IMDb()
```

Slika 3.4: Vključitev knjižnice IMDb in vzpostavitev povezave

### Iskanje objektov

Za iskanje so v knjižnici implementirane funkcije, ki poleg iskanega niza vrnejo seznam objektov tudi s podobnimi rezultati. Primer iskanja filma z imenom "matrix" je prikazan na sliki 3.5. Podobno lahko iščemo po imenih igralcev z uporabo funkcije `ia.search_person('Actor Name')` [2].

### Podatki objekta

Rezultati iskanja so objekti, ki se obnašajo kot slovarji v obliki ključev (key) in vrednosti (value). Vsak objekt ima definirane prevzete vrednosti, ki jih

```
>>> movies = ia.search_movie('matrix')
>>> movies[0]
<Movie id:0133093[http] title:_The Matrix (1999)_>
```

Slika 3.5: Iskanje filma

izpišemo z ukazom `Objekt.default_info`. Vse ključe, ki jih vsebuje objekt, poiščemo z ukazom `Objekt.infoset2keys` (glej sliko 3.6).

```
>>> movie = ia.get_movie('0133093')
>>> movie.infoset2keys
{'main': ['cast', 'genres', ..., 'top 250 rank'], 'plot': ['plot', 'synopsis']}
```

Slika 3.6: Izpis vseh ključev objekta

### Rezultati iskanja vsebujejo podobna imena

Funkcija v primeru iskanja imena Aleks vrne tudi igralce z imenom Aleksander. Za potrebe naše aplikacije so takšni rezultati nezaželjeni. Želimo le igralce, katerih ime se natanko ujema z iskanim. Težavo rešimo s filtriranjem rezultatov.

Objekt tipa igralec (actor) vsebuje polje ime (name) katerega smo uporabili za filtriranje odvečnih igralcev. Ime igralca pridobimo z ukazom `person['name']`. Žal ukaz vrne ime in priimek v obliki besedila. Samo ime izluščimo tako, da celotno besedilo razdelimo na mestu presledka. Dobljeno vrednost primerjamo z željenim imenom in v primeru enakosti z ukazom `oseba.get("filmography")` pridobimo seznam filmov.

### Struktura in pošiljanje podatkov

Za namen spletne aplikacije sta uporabljeni le dve izmed vseh vrednosti, ki jih vsebuje objekt tipa film. Strežnik odgovarja na poizvedbe v HTTP-protokolu. Vsi podatki se pošiljajo v telesu odgovora (message-body) v JSON-obliki (glej sliko 3.7). Odgovor zgradimo z uporabo dveh zank: prva


Slika 3.7: Primer odgovora strežnika IMDbPY

se sprehodi skozi vse igralce, druga skozi seznam vseh filmov igralca in izlušči podatke za ime in leto filma.

## 3.3 Prosta spletna enciklopedija Wikipedija

### 3.3.1 Podatki Wikipedije

Wikipedija je prosta spletna enciklopedija v več jezikih. Na spletni strani so zbrani mnogi članki za najrazličnejše kategorije. Za potrebo diplomske naloge smo se osredotočili zgolj na članke osebnih imen.

Podatke za iskano ime dobimo tako, da obiščemo glavno stran Wikipedije in v iskalni niz vnesemo iskano ime. V primeru obstoja le enega članka na iskani niz se podatki samodejno izpišejo. V nekaterih primerih, kot je ime Andi, naletimo na težavo, saj poleg osebnega imena obstaja tudi gorovje z identičnim imenom. V tem primeru spletna stran Wikipedije namesto željenih podatkov za iskano ime Andi vrne podatke gorovja z identičnim imenom.

Podatki na spletni strani so v obliki spletne datoteke HTML. Spletna stran članka je sestavljena iz standardnih HTML5-značk (glava (`<head>`),

navigacija (`<nav>`), telo (`<body>`) in noga (`<footer>`)). Vsi podatki o imenu, ki nas zanimajo, se nahajajo v znački telo (`<body>`). Značka vsebuje več odstavkov in eno tabelo. Med vsemi podatki izluščimo podatke, tabele in najzanimivejše odstavke, kot so osebni praznik, spol, god, zanimivosti, pomen itn. Nekatere nerelavantne odstavke smo izpustili.

# Poglavje 4

# Dostop do različnih spletnih virov

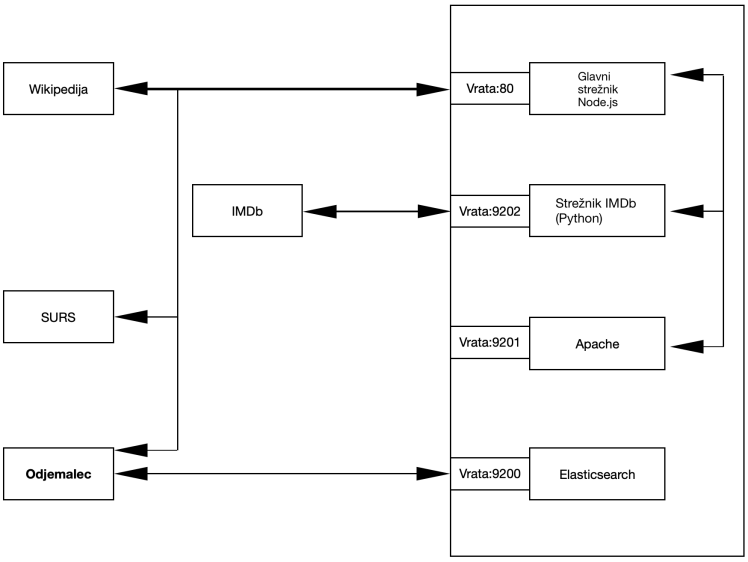
Za delovanje spretne strani potrebujemo dostop do različnih virov (Wikipedije, IMDb, SURS), ki so dostopni na različnih spletnih naslovih. Zato se pojavi težava navzkrižne deliteve podatkov (CORS) [11].

Težavi se izognemo z uporabo le enega strežnika, ki preusmerja zahtevke na vse ostale. Zato je potrebno določiti spletne naslove in vrata (port), na katerih strežniki poslušajo in odgovarjajo na zahtevke. Celotna arhitektura je prikazana na sliki 4.1.

## 4.1 Navzkrižna delitev podatkov (CORS)

Razlog za opisano arhitekturo je v varnosti. Brskalnik za prikaz spletne strani potrebuje datoteke, kot so HTML-datoteke, slike, kaskadne stilske predloge, JavaScript datoteke itn. Vse vrste datotek pridobi z zahtevami po protokolu HTTP. Običajno se vse datoteke spletne strani nahajajo na enem strežniku. Za lažjo predstavitev vzemimo primer dostopa do privzete spletne strani, imenovane index.html, na naslovu 52.233.168.221. HTTP-zahteva je v tem primeru sledeča:

```
GET 52.233.168.221/index.html HTTP/1.1
```




Slika 4.1: Nastavitev vrat in komunikacija med strežniki

```
User-Agent:  Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host:  52.233.168.221:80
Accept-Language:  en-us
Accept-Encoding:  gzip, deflate
Connection:  Keep-Alive
```

Prva vrstica predstavlja vrsto, naslov in datoteko. Naslov je v tem primeru `52.233.168.221`. Na zahtevo po spletni strani strežnik odgovori z odgovorom, v katerem je vsebina zahtevane datoteke. Brskalnik prebere vsebino datoteke in izlušči nove povezave oz. datoteke, ki jih potrebuje za prikaz. Za primer vzemimo, da potrebuje tri datoteke z naslovi:

- `52.233.168.221/style.css`
- `52.233.168.221:9202/queryimdb/acror/Aleks`
- `www.wikipedija.com/Aleks`

Brskalnik pošlje poizvedbo po vseh treh spletnih straneh, a bo blokiral

vsebino spletnima stranema dve in tri. Prvo spletno stran bo obravnaval brez težav, saj ima enako lokacijo kot pred tem iskana datoteka index.html (prva zahteva nima posebej navedenih vrat, zato se privzame številka 80). Drugo spletno stran bo blokiral zaradi razlike v številki vrat (9202). Tretja zahteva ima popolnoma drugačen naslov.

Težava mehanizma CORS se pojavi v primeru, ko se naslovi strežnika med seboj razlikujejo. V tem primeru brskalnik zazna različne lokacije HTTP-zahteve in vsebino blokira. Vsebino blokira z razlogom, saj bi bila vsebina datoteke iz drugega vira lahko zlonamerna.

## 4.2 Elasticsearch

Elasticsearch bo kot edini neposredno (brez glavnega strežnika) odgovarjal na poizvedbe na prvih javno dostopnih vratih (port) s številko 9200. Mehanizem CORS ga zaradi spodaj navedenih nastavitev ne bo blokiral.

Prevzeta številka vrat za Elasticsearch je 9200, kar je natanko to, kar potrebujemo. A prevzete nastavitve v konfiguracijski datoteki `elasticsearch.yml` ne omogočajo dostopa do podatkov, razen strežniku samemu (localhost). Za dostop zunaj strežnika dodamo nekaj spodaj navedenih vrstic v konfiguracijsko datoteko `elasticsearch.yml`:

```
http.cors.enabled: true
http.cors.allow-origin:
network.host: 0.0.0.0.
```

Nastavitve naredijo sledeče: preprečijo težavo mehanizma CORS in omogočijo javen dostop do podatkovne zbirke.

## 4.3 Strežnik IMDb

Strežnik IMDb, ki streže podatke podatkovne zbirke IMDb in je napisan v programskem jeziku Python, bo odgovarjal na vratih (portu) številka 9202, a le na zahtevke, katerih izvor bo fizični strežnik sam (localhost). Javni dostop

do strežnika ne bo mogoč. Strežnik sicer nima nikakršnih omejitev glede odgovorov na zahtevke izven lokalnega omrežja. To pomeni, da če bi imeli na razpolago še ena odprta vrata, npr. 9010, in bi strežnik stregel na tem naslovu, bi z ustrezno poizvedbo brez težav dobili odgovor.

## 4.4 Apache

Apache je najbolj priljubljen odprtokodni strežnik, ki prevzeto teče na vratih številka 80 in je namenjen streženju spletnih strani. Številko vrat smo za namen diplomske naloge prenastavili na vrata številka `9201`. S tem strežnik postane nedosegljiv iz zunanje mreže. Številko vrat za Apache nastavimo v datoteki `/etc/apache2/sites-enabled/000-default.conf`. V konfiguraciji nastavimo vrstico `Listen [stevilka_vrat]`. Nastavitve shranimo in ponovno zaženemo streznik.

## 4.5 Glavni strežnik Node.js

Zaradi mehanizma CORS imajo vsi zahtevki, ki jih spletna aplikacija potrebuje za delovanje, enak naslov: 52.233.168.221, a se bodo med seboj razlikovali v glavi HTTP-zahtevka.

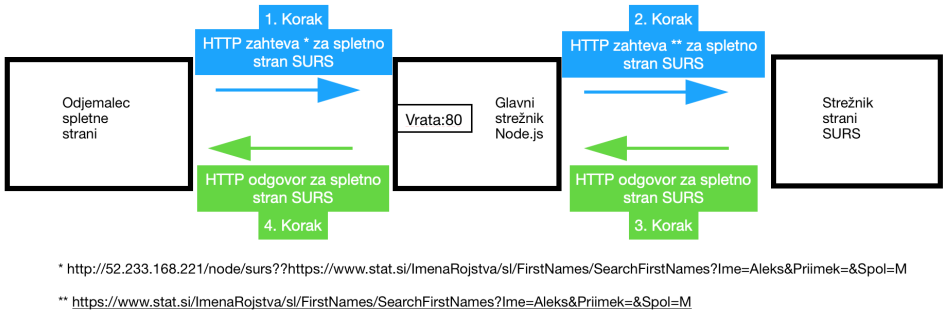
Glavni vir aplikacije je spletni strežnik, napisan v Node.js. Na vratih številka 80 odgovarja na vse zahteve z izjemo podatkov, ki so shranjeni v podatkovni zbirki Elasticsearch. Da smo še bolj natančni, bo neposredno odgovarjal le na poizvedbe po spletni strani Wikipedije in spletni strani SURS. Ostali zahtevki so preusmerjeni na strežnike, omenjene v prejšnjem poglavju. Pojavi se sledeče vprašanje: "Kako bomo ločili zahtevke, glede na kateri strežnik morajo biti preusmerjeni?" Odgovor je sledeč: "Vse poizvedbe podatkov potekajo preko protokola HTTP."

HTTP je protokol, ki je namenjen prenosu spletnih datotek. Zahtevek je sestavljen iz 4 delov: naslova spletne strani, glave zahteve, prazne vrstice in vsebine. Naslov spletne strani je v našem primeru uporabljen za določitev,

na kateri spletni strežnik se preusmeri HTTP-zahteva.

### Zahtevki na spletno stran SURS

V primeru zahtevka po podatkih, ki so dostopni na spletni strani SURS-a za skupno število imen, se izvede poizvedba preko glavnega strežnika Node.js. Vsi zahtevki odjemalca morajo imeti v glavi za naslovom predpono `surs`, čemur sledi znak "?" in spletna stran SURS-a. Za primer poizvedbe po imenu Aleks je naslov sledeč:
`http:52.233.168.221/node/surs??https://www.stat.si/ImenaRojstva/sl/FirstNames/SearchFirstNames?Ime=Aleks&Priimek=&Spol=M`. Glavni strežnik v tem primeru izvede neposredno poizvedbo na spletno stran SURS-a in vrne podatke odjemalcu (glej sliko 4.2).

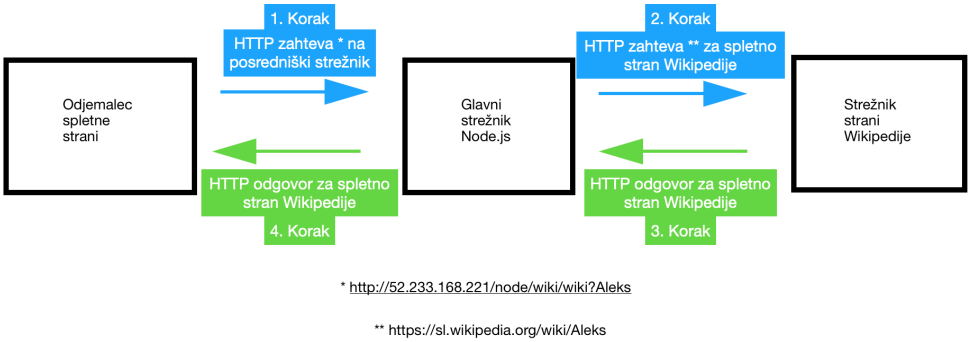



Slika 4.2: Komunikacija med odjemalcem, glavnim strežnikom in spletno stranjo SURS

### Zahtevki na spletno stran Wikipedije

Vsi zahtevki na spletno stran Wikipedije se obravnavajo preko glavnega strežnika, napisanega v programskem jeziku Node.js, in imajo v glavi HTTP-zahtevka predpono `node` in `wiki`. Primer zahtevka za ime Aleks je:
`http://52.233.168.221/node/wiki/wiki?Aleks`. Strežnik v primeru te

vrste zahtevka izvede poizvedbo na spletno stran Wikipedije na ustrezen naslov in njen odgovor posreduje odjemalcu (glej sliko 4.3).



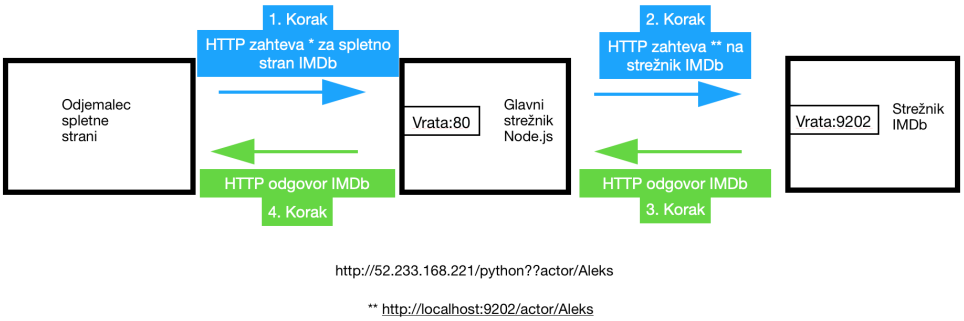

Slika 4.3: Komunikacija med odjemalcem, glavnim strežnikom in spletno stranjo Wikipedije

### Zahtevki po podatkih IMDb

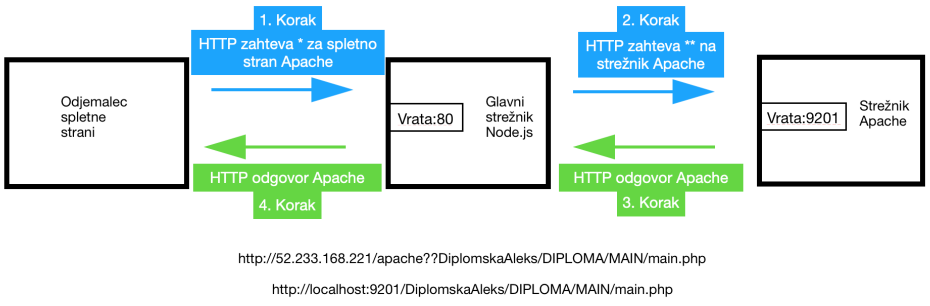
Zahtevki po podatkih o igralcih in filmih bodo preko glavnega strežnika preusmerjeni na strežnik IMDb (opisan v poglavju 4.3). Identificirani bodo na podlagi naslova s predpono `python`. Za primer iskanja igralcev z imenom Aleks naslov poizvedbe izgleda sledeče: `http //52.233.168.221/python??actor/Aleks`. Glavni strežnik preusmeri poizvedbo na naslov: `http://localhost:9202/actor/Aleks` (glej sliko 4.4).

### Zahtevki spletnih strani Apache

Vrata strežnika Apache smo prenastavili, zato moramo vse zahtevke spletne aplikacije preusmeriti na vrata 9201. Tudi tokrat bomo uporabili glavni strežnik. Vsi zahtevki morajo vsebovati naslov `52.233.168.221`, predpono `apache??` in pot oz. datoteko, ki jo zahtevajo. Primer zahteve po glavni spletni strani aplikacije je: `http //52.233.168.221/apache??DiplomskaAleks/DIPLOMA/MAIN/main.php` (glej sliko 4.5).




Slika 4.4: Komunikacija med odjemalcem, glavnim strežnikom in strežnikom IMDb




Slika 4.5: Komunikacija med odjemalcem, glavnim strežnikom in strežnikom Apache

# Poglavje 5

# Opis aplikacije

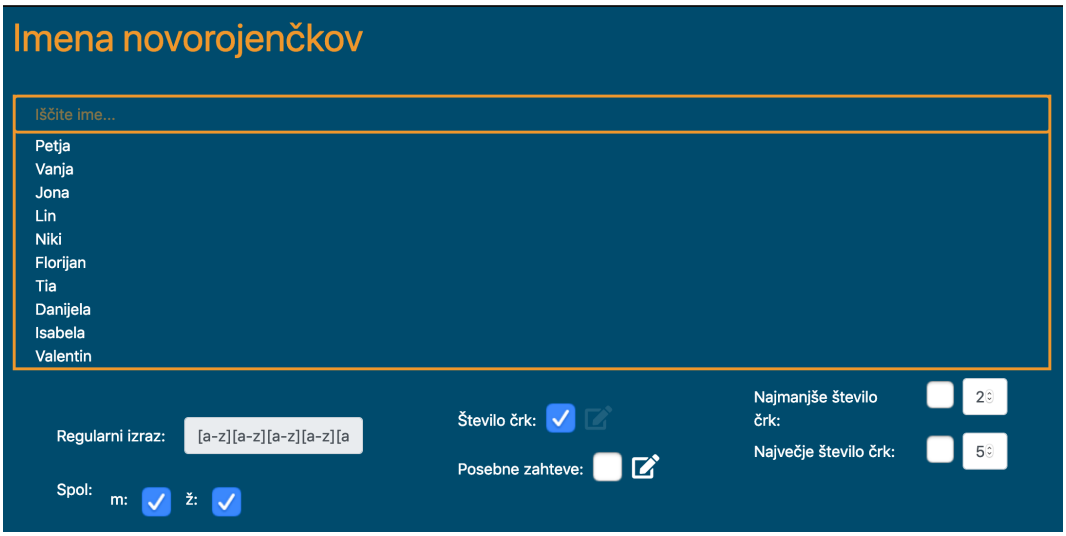
Spletno aplikacijo bomo, zaradi lažjega razumevanja, razdelili na štiri dele:

1. Polje za iskanje in filtriranje.
2. Graf vizualizacije podatkov o številu imen skozi čas.
3. Graf filmskih igralcev.
4. Podrobnosti imena.

## 5.1 Polje za iskanje in filtriranje imen

Zgornji del aplikacije je namenjen iskanju in filtriranju imen (glej sliko 5.1). V prvo vnosno polje uporabnik vnese iskano ime ali prve črke imena. Z vsakim vnosom črke v vnosno polje se pod njim osvežijo in prikažejo predlogi imen. Seznam predlogov lahko omejimo z uporabo filtrov. Prvi filter omejuje imena po spolu. Prvo potrditveno polje prikaže le imena dečkov, drugo imena deklic.

Srednji del sestavljata dva filtra: "Število črk" in "Posebne zahteve". Posamezno vrsto izberemo s klikom na potrditveno polje. Z izbiro filtra "Število črk" se na skrajno desni strani prikaže nastavitev dveh podfiltrov. Zgornji "Najmanjše število črk" omejuje najmanjše število črk imena. Spodnji "Največje število črk" omejuje na enak način, le da omejuje največje število




Slika 5.1: Polje za iskanje in filtriranje

črk imena. S klikom na puščici v smer gor ali dol določimo minimalno in maksimalno število črk. V primeru, da vrednost minimalnega število črk preseže maksimalnega, se vrednost za maksimalno število črk samodejno poveča. Posamezno omejitev za minimalno ali maksimalno omejitev je potrebno potrditi z izbiro potrditvenega polja.

Spodnji filter z imenom "Posebne zahteve" omogoča iskanje imen glede na črke, ki jih ime vsebuje ali ne vsebuje na posameznem mestu. Z izbiro filtra "Število črk" se samodejno izbere tudi filter za omejitev števila črk. Nabor črk na poljubnem mestu omejimo z izbiro krogca, ki vsebuje številko mesta, na katerem želimo omejiti črke. Spodnji "Največje število črk" omejuje na enak način, le da omejuje največje število črk imena. S klikom na potrditveno polje "Vsebuje/ne vsebuje črke" izberemo, ali črke želimo na tem mestu ali ne. Neizbrane črke so modre barve. Neželjene izbrane črke se obarvajo rdeče. Željene izbrane črke se obarvajo zeleno. Ob vsaki spremembi filtra se osveži vrednost polja regularnega izraza. Na podlagi generiranega regularnega izraza se seznam predlogov samodejno posodobi. V danem trenutku je na seznamu predlogov največ 10 imen oz. vsa imena, ki izpolnjujejo

pogoje nastavljenega filtra. Filter odstranimo s preprostim klikom na vnosno polje. Nastavljeni filter skrijemo ali prikažemo s klikom na gumb za urejanje (glej sliko 5.2).


Slika 5.2: Filtriranje črk imena

## 5.2 Vizualizacija števila imen novorojenčnkov

### 5.2.1 Pridobivanje podatkov na strani odjemalca

Imena, ki jih želimo prikazati na grafu, so shranjena v obliki seznama. Na podlagi seznama se generira regularni izraz. Ta je sestavljen iz vseh aktivnih imen s seznama. Med imeni je beseda OR. Za primer vzemimo seznam, ki

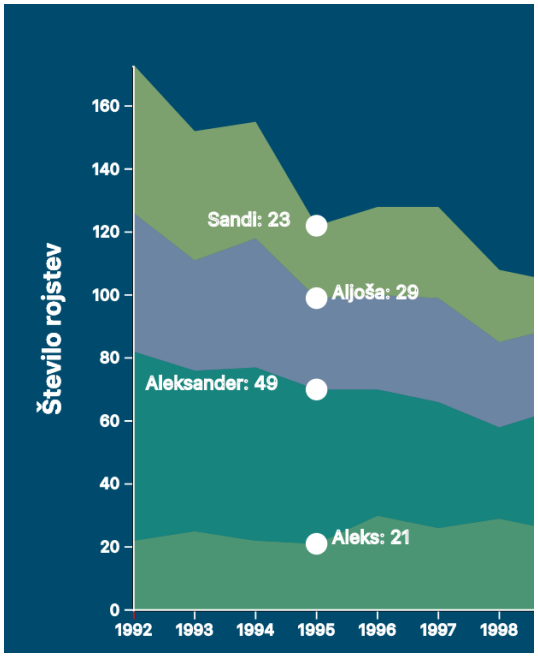
vsebuje imena: Aleks, Aleksander, Aljoša in Sandi. Regularni izraz za seznam je torej `Aleks OR Aleksander OR Aljoša OR Sandi`. Regularni izraz se kot parameter pošlje v polje telesa HTTP-zahtevka. Odgovor strežnika Elasticsearch je v obliki JSON in vsebuje slovar, v katerem so vsa iskana imena in vrednosti za posamezno leto.

### 5.2.2 Prikaz podatkov

Za vizualizacijo pojavitev imen novorojenčkov skozi čas sem izbral naložen ploščinski graf, ki prikazuje število pojavitev od leta 1992 naprej za vsa iskana imena. Os x predstavlja čas oziroma leta od 1992 do 2017. Os y predstavlja število pojavitev. Za izris grafa je uporabljena knjižnica D3.js, ki omogoča izrisovanje različnih vrst grafov, vse od osnovnih do bolj zapletenih. Izrisovanje je poenostavljeno, a še vedno zahteva kar nekaj znanja v primerjavi s knjižnico Plotly [7]. Podatki za izris morajo imeti predpisano strukturo. Za izris osi definiramo minimalno in maksimalno število, tako za letnice kot za število pojavitev.

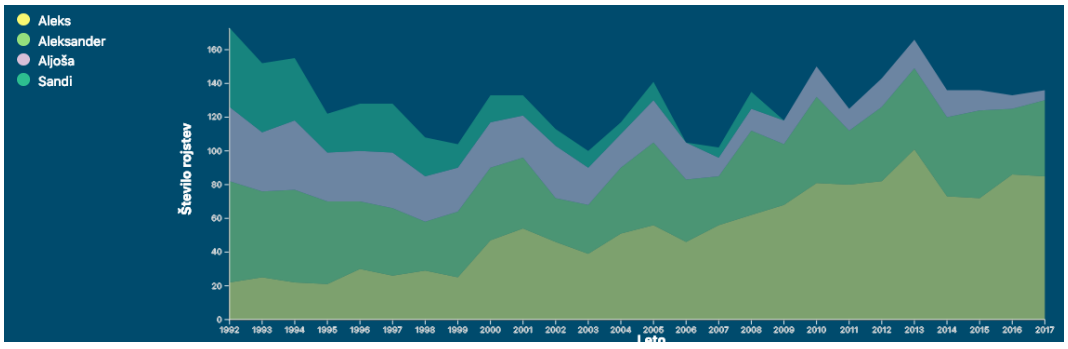
V primeru, ko imamo na grafu več imen, zgornja vrednost predstavlja seštevek vseh imen pod njim za določeno leto.

Število pojavitev vseh imen za določeno leto lahko vidimo, če se z miško postavimo na željeno leto. Vse vrednosti se prikažejo levo ali desno ob novo narisanem krogu (glej sliko 5.3). Idejo za tako predstavitev sem dobil na spletni strani [17, 18]. Krogce, ki se izrišejo, ko se z miško pomaknemo na leto, sem implementiral sam. Funkcija za izris se sprehodi po seznamu imen in za vsako leto prebere vrednost. Na podlagi vrednosti, robov in vrednosti prejšnjega imena izračuna pozicijo kroga. Sledi izpis imena, ki ga predstavlja krogec in sama vrednost, ki predstavlja število rojstev s tem imenom. Pozicija besedila je lahko na levi ali na desni strani, odvisno od zaporedne številke (levo so pozicijonirane vrednosti na lihi poziciji, desno so pozicijonirane vrednosti na sodi poziciji). Poziciji levo in desno se izmenjujeta, da preprečimo prekrivanje, do katerega lahko privede nizko število pojavitev.




Slika 5.3: Graf - število imen novorojenčkov za določeno leto

Ob grafu se nahaja legenda (glej sliko 5.4). Legenda prikazuje zgodovino iskanja imen in omogoča manipulacijo imen, ki so prikazana na grafu. Imena ob krogih z barvno vsebino so prikazana na grafu. Imena lahko odstranimo z grafa s klikom na krog ob imenu. Imena, katerih krogi nimajo barvne vsebine, niso prikazani na grafu.




Slika 5.4: Graf – število imen novorojenčkov skozi čas

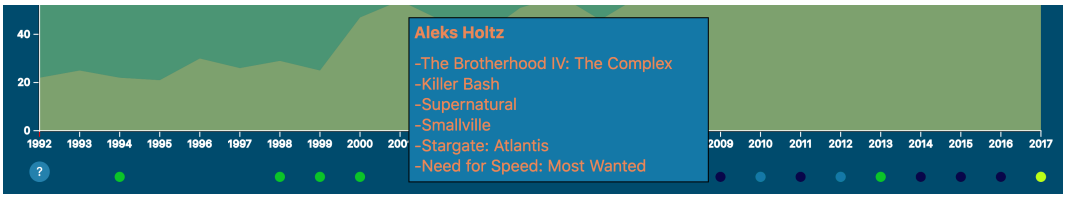
#### Imena, ki so tako ženska kot moška

Med podatki obstajajo imena, ki so lahko moška ali ženska. Ta imena, ob nezavedanju, na grafu niso prikazana. Poizvedba podatkov za te vrste imen se izvede brez težav, a pri obdelavi podatkov oz. izrisovanju napaka postane resna do te mere, da se na graf ne izrišejo vrednosti. Razlog sta dve vrednosti za vsako leto, kar pri seštevanju vrednosti povzroči napako.

Težavo zaznamo tako, da preštejemo, kolikokrat se v zapisu pojavi ena letnica. V primeru, ko se pojavi več kot enkrat, vemo, da je ime lahko tako moško kot žensko. Težavo se da odpraviti na več načinov. Prva rešitev bi bila združitev in prikaz le ene površine. Druga rešitev, ki je uporabljena v diplomski nalogi, je izris imena ločeno glede na spol, kar pomeni, da najprej ločimo podatke za žensko in moško ime in izrišemo dve ločeni površini.

## 5.3 Vizualizacija podatkov IMDb

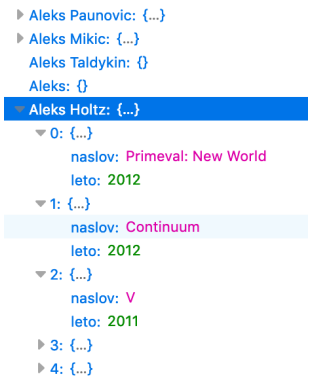
Pod naloženim grafom se nahaja vizualizacija podatkov iz podatkovne baze IMDb (glej sliko 5.5). Podatki prikazujejo filmske igralce le za zadnje iskano ime. Vsak igralec je predstavljen z naključno barvo, kar omogoča hitrejše iskanje in povezovanje igralcev po letih igranja. Vsak krogec predstavlja enega igralca in vse njegove filme za izbrano leto. Podatke lahko vidimo, če se z miško premaknemo na enega od krogcev. Nad njim se prikaže kvadrat v enaki barvi. V njem se v prvi vrstici prikaže ime igralca, pod njim pa seznam igranih filmov.




Slika 5.5: Graf – Imena igralcev in filmov

### 5.3.1 Pridobivanje podatkov

Skripta na strani odjemalca izvede HTTP-poizvedbo na naslov, npr.: `http:52.233.168.221/python??actor/Aleks`. Za pošiljanje je uporabljen standard AJAX in objekt `XMLHttpRequest`. Funkcija, ki izvede HTTP-poizvedbo, je asinhrona, kar pomeni, da je potrebno paziti, saj se naslednji ukaz (prikaz podatkov) lahko izvede vzporedno. Zato moramo poskrbeti, da odgovor na poizvedbo dobimo, preden nadaljujemo z obdelavo podatkov. Objekt XMLHttpRequest vsebuje spremenljivko, v kateri je shranjeno stanje poizvedbe. V primeru prejetega odgovora ima spremenljivka vrednost 4. HTTP-protokol poleg ostalih podatkov, ki jih pošlje, vsebuje tudi status. V primeru uspešnega odgovora ima številko 200. Rezultati poizvedbe so v obliki JSON-objekta. Ključ predstavja ime igralca. Vrednosti predstavlja seznam filmov v JSON-obliki (glej sliko 5.6).




Slika 5.6: Odgovor IMDb-strežnika

### 5.3.2 Prikaz podatkov

Koraki prikaza podatkov so naslednji:

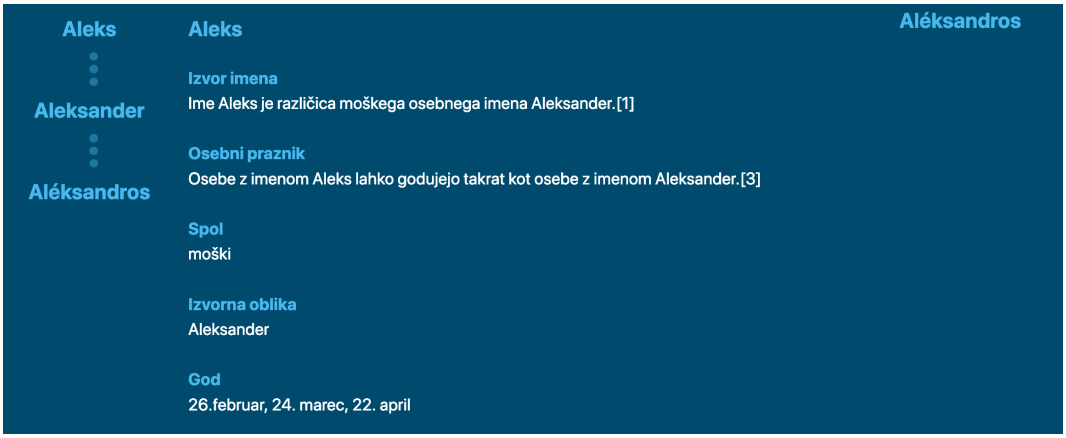
1. Sprehodimo se skozi seznam igralcev.

2. Filme združimo po letih.

3. Za vsakega igralca in vsako leto izrišemo en krog. Pri tem je potrebno za vsak krog, ki ga dodamo za določeno leto, izračunati pozicijo y, da preprečimo prekrivanje pred tem izrisanih krogov.

## 5.4 Vizualizacija podatkov Wikipedije

Zadnji spodnji del vsebuje zanimive podatke o imenu (glej sliko 5.7). Vir podatkov predstavljata Wikipedija in spletna stran SURS. Med vsemi podatki sem izluščil le nekatere, ki so na strani prikazani z identično strukturo Wikipediji, v obliki naslova in besedila. Ime, za katerega so podatki izpisani, je prikazano v svetlo modri barvi, z večjo pisavo in odebeljeno. Vsi naslovi so identične oblike kot ime, le velikost besedila je manjša.




Slika 5.7: Podatki o imenu

### 5.4.1 Prikaz podatkov iz Wikipedije

Prikaz podatkov iz Wikipedije je bolj zahteven, kot izgleda na prvi pogled. Koraki za prikaz podatkov so:

1. Ustvari prazen seznam S=[].

2. Pridobi spletno stran Wikipedije za iskano ime, npr: Aleks s poizvedbo na glavni strežnik aplikacije na naslov.:
`http://52.233.168.321/node/wiki/wiki?Aleks`

3. Preveri, ali spletna stran vsebuje podatke za ime (pazimo na imena, kot so Andi).

   (a) V primeru, da je spletna stran za ime X, nadaljuj na korak 4.

   (b) V primeru, da spletna stran ni namenjena imenu, nadaljuj na korak 5.

4. Preveri, ali izvorna oblika obstaja:

   (a) Izpiši osnovne podatke za iskano ime v obliki naslova in besedila.

   (b) V primeru, da izvorna oblika obstaja, shrani ime kot Y. Dodaj Y v seznam S in izvedi korake od 1 in 3 z imenom Y.

   (c) V primeru, da izvorna oblika ne obstaja, zaključi izvajanje korakov in vrni seznam S.

5. Novo ime X = X + "_(ime)". Vrni se na prvi korak.

### Problematična imena

Nekatera imena, kot so npr. Andi, so povzročala obilico težav, saj skripta deluje na osnovi povezave URL. Povezava URL do imena ima običajno sledečo strukturo: `https://sl.wikipedia.org/wiki/Aleksander`. Zadnja beseda v tem primeru Aleksander predstavlja ime, ki ga iščemo. A v posebnih primerih, kot je Andi, dobimo vsebino, ki se nananša na gorovje. Težavo lahko rešimo tako, da pridobljeno vsebino spletne strani pregledamo in poiščemo specifične naslove, značilne za ime. V primeru napačne vsebine spletne strani izvedemo novo poizvedbo. Na koncu zahtevanega naslova za iskanim imenom dodamo *_ime*. Primer naslova:
`http 52.233.168.221/node/wiki?Andi_(ime)`.

#### Izvorna oblika imena

Nekatera imena imajo kot predhodnika navedenega naslednika. Npr. ime X ima kot prednika navedeno ime Y. Ime Y ima kot prednika navedeno ime X. Kar pomeni, da v primeru neprevidnosti hitro povzročimo neskončno zanko. Rešitev je preprosta: vsako ime, ki smo ga našli kot prednika, dodamo v seznam. Preden izvedemo rekurzijo za novo ime, preverimo, ali se že nahaja v seznamu, in v tem primeru rekurzijo zaključimo.

Imena na levi strani prikazujejo izvor imena (glej sliko 5.8). Izvorna oblika imena je prikazana pod iskanim imenom. Zadnje ime na seznamu nima predhodnika oz. ni znan. Imena so ločena s krogi.


Slika 5.8: Prikaz izvora imena

### 5.4.2 Prikaz podatkov s spletne strani SURS

### 5.4.3 Pridobivanje podatkov

Podatke, omenjene v poglavju 3.1.2, s Statističnega urada Republike Slovenije, pridobimo na identičen način kot podatke Wikipedije. Razlika je le v naslovu HTTP-zahtevka, ki je v tem primeru: `http //52.233.168.221/node/surs?? https www.stat.si/ImenaRojstva/sl/FirstNames/SearchFirstNames?Im e=Aleks&Priimek=&Spol=M`

### 5.4.4 Prikaz podatkov

Rezultat poizvedbe je vsebina spletne strani Statističnega urada Republike Slovenije. Skripta na strani odjemalca le izlušči odstavek, predstavljen v

poglavju 3.1.2, in ga prikaže z identično obliko, kot smo pred tem prikazali podatke spletne enciklopedije Wikipedija (glej sliko 5.9).


Slika 5.9: Prikaz podatkov Statističnega urada Republike Slovenije za daljše obdobje

# Poglavje 6

# Povzetek

Razvita je aplikacija, ki omogoča več kot le iskanje podatkov o številu imen novorojenčkov. Dosegli smo cilj in aplikacijo približali uporabniku, jo naredili atraktivno in uporabniku olajšali iskanje in izbiro imena za svojega otroka.

Aplikacijo bi lahko izboljšali praktično na vseh področjih. Uporabniku bi lahko prikazali podatke iz še več različnih virov. Mogoče bi bilo zanimivo primerjati slovenska imena z imeni sosednjih držav in odkrivati, ali obstajajo povezave.

Aplikacija je zaradi velike količine podatkov oz. poizvedb, ki jih potrebuje, zahtevna za omrežje. Ena izmed rešitev bi bila uporaba Kafke.

Med izdelavo aplikacije je prišlo do več težav, ki jih pred začetkom nismo predvideli. Najbolj pomembne so predstavljene v diplomski nalogi. Ena najbolj pomembnih je nedvomno mehanizem CORS, a obenem je tudi najbolj poučna. Med izdelavo sem pridobil nova znanja o implementaciji strežnikov. Naučil pa sem se tudi različnih načinov upravljanja z asinhronim delovanjem JavaScripta.

# Literatura


[1] D3.js. Dosegljivo: https://d3js.org/. [Dostopano: 24. 2. 2019].

[2] Dokumentacija knjižnice IMDbPY. Dosegljivo: https://media.readthedocs.org/pdf/imdbpy/stable/imdbpy.pdf. [Dostopano: 24. 2. 2019].

[3] Elasticsearch. Dosegljivo: https://www.elastic.co/. [Dostopano: 24. 2. 2019].

[4] HTTP protokol. Dosegljivo: https://tools.ietf.org/html/rfc2616. [Dostopano: 27. 2. 2019].

[5] IMDb podatkovna zbirka. Dosegljivo: https://datasets.imdbws.com/. [Dostopano: 24. 2. 2019].

[6] JavaScript. Dosegljivo: https://www.w3schools.com/js/. [Dostopano: 24. 2. 2019].

[7] Knjižnica Plotly. Dosegljivo: https://plot.ly/feed/#/. [Dostopano: 28. 2. 2019].

[8] Linux. Dosegljivo: https://www.linux.org/. [Dostopano: 24. 2. 2019].

[9] Nastavitev okoljskih spremenljivk. Dosegljivo: https://gerardnico.com/os/linux/locale. [Dostopano: 25. 2. 2019].

[10] Nastavitev okoljskih spremenljivk z ukazom export. Dosegljivo: https://www.cyberciti.biz/faq/linux-unix-shell-export-command/. [Dostopano: 25. 2. 2019].

[11] Navzkrižna delitev podatkov. Dosegljivo: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS. [Dostopano: 24. 2. 2019].

[12] Node.js. Dosegljivo: https://nodejs.org/en/. [Dostopano: 24. 2. 2019].

[13] Node.js - W3 schools. Dosegljivo: https://www.w3schools.com/nodejs/nodejs_intro.asp. [Dostopano: 25. 2. 2019].

[14] Podatki Statističnega urada Republike Slovenije za deklice. Dosegljivo: https://pxweb.stat.si/pxweb/Dialog/varval.asp?ma=05X2002S&ti=&path=../database/Dem_soc/05_prebivalstvo/46_Imena_priimki/07_05X20_imena_novoroj/&lang=2. [Dostopano: 25. 2. 2019].

[15] Podatki Statističnega urada Republike Slovenije za dečke. Dosegljivo: https://pxweb.stat.si/pxweb/Dialog/varval.asp?ma=05X2001S&ti=&path=../database/Dem_soc/05_prebivalstvo/46_Imena_priimki/07_05X20_imena_novoroj/&lang=2. [Dostopano: 25. 2. 2019].

[16] Podatkovna zbirka IMDb. Dosegljivo: https://www.imdb.com/interfaces/. [Dostopano: 25. 2. 2019].

[17] Primer naloženega grafa. Dosegljivo: http://bl.ocks.org/anaeliaovalle/e57763e85def2a95be931c69eff6bfa6. [Dostopano: 24. 2. 2019].

[18] Primer napisov na grafu. Dosegljivo: https://bl.ocks.org/fabiomainardi/3976176cb36e718a608f. [Dostopano: 24. 2. 2019].

[19] Primer vizualizacije SURS. Dosegljivo: https://stat.si/womenmen/bloc-2a.html?lang=sl. [Dostopano: 24. 2. 2019].

[20] Python. Dosegljivo: https://www.python.org/. [Dostopano: 24. 2. 2019].

[21] Python knjižnica za datoteke tipa CSV. Dosegljivo: https://docs.python.org/3/library/csv.html. [Dostopano: 25. 2. 2019].

[22] Python knjižnica za Elasticsearch. Dosegljivo: https://elasticsearch-py.readthedocs.io/en/master/. [Dostopano: 25. 2. 2019].

[23] Spletna enciklopedija Wikipedija. Dosegljivo: https://sl.wikipedia.org/wiki/Glavna_stran. [Dostopano: 24. 2. 2019].

[24] Spletna stran IMDb. Dosegljivo: https://help.imdb.com/article/imdb/general-information/what-is-imdb/G836CY29Z4SGNMK5?ref_=__seemr#. [Dostopano: 25. 2. 2019].